Documentation

Header File

tic.h:

Class tBoard:

Meant to be the class that handles the creation of the tic-tac-toe board, as well as the players moves, and the winner of the game after the game has been found to of ended.

Private attributes:

there are no private attributes.

Public Attributes:

vector <symbol> tac;

Declaration of a vector attribute to hold all the possible spaces there are for a player to play a piece, or in this case a “symbol”.

tBoard::tBoard();

Default constructor for the tic-tac-board, it creates the data needed to output the board to the user and is generated having initially only “BLANK” spaces, a predeclared symbol for the program to use.

Bool tBoard::move(symbol m, int x, int y);

Used to implement a move input from a player, using m as the symbol that the user uses (being “X” or “O” in this case) to be placed in a spot on the board indicated by the “x” and “y” values passed to the function. As long as there is not already a symbol other than “BLANK” in the spot indicated, the move is played and the function returns true. If the move is not a legal move, then the function returns false.

Bool tBoard::game_over();

Checks if the game has concluded in a win of a player or a tie, if the game is over it returns true, otherwise the function return false.

Symbol tBoard:: winner();

Checks if the game has concluded in a win of a player or a tie, and returns the symbol of the winner or a “BLANK” symbol if it ended in a tie.

Bool tBoard::areSame(int x, int y, int z);

Checks if the three symbols found at the index’s of the tBoard object’s vector indicated by the three ints passed to the function, are the same symbols.

ostream & operator<<(ostream& os, const tBoard& myTable);

Declared so that the tBoard object can be outputted for the user to see the board.

Implementation File

tic.cpp:

ostream& operator<<(ostream& os, const tBoard& myTable):

Using the tic-tac-toe board formatting provided by the lab and the symbols included in the symbol header file, this operator is overloaded to simply output the board with each of the elements from the tBoard's vector into their rightful spots starting with the top left being the 0th element and the bottom right being the 8th element.

tBoard::tBoard()

This constructor function simply created the object with a vector of nine "BLANK" symbols using a for loop.

bool tBoard::move(symbol m, int x, int y)

A string is declared as "spotString" and is assigned the ints passed to the function "x" and "y" casted as strings and combined together. This string is then made into an int using stoi(spotString) and is set to an int variable named spot. Then another int variable named element is declared to hold data later on. A switch statement is then implemented to see if the int "spot" is equal to a list of 9 different ints going from 00 to 22 in base 2 math (I think). As a result of which number spot is equal to, a number 0 through 8 is assigned to element. Element is then used as an index of the tBoard object's vector to see if the symbol in that index is a "BLANK" or not. If the symbol found in the index is a blank, then the current user's symbol is placed in the index and "true" is returned, otherwise "false" is returned.

bool tBoard::game_over()

An if-else if stream of statements is declared checking all possible outcomes of a finished game. In the case of a tie, the board is checked to see if all the spaces on the board have been filled after checking for all patterns of three in a row. If there is a winner or a tie, “true” is returned, otherwise “false” is returned.

symbol tBoard::winner()

An if-else if stream of statements is declared checking all possible outcomes of a finished game and the symbol found to of won is returned. In the case of a tie, the board is checked to see if all the spaces on the board have been filled after checking for all patterns of three in a row. If there is a tie, “BLANK” is returned.

bool tBoard::areSame(int x, int y, int z)

An if statement first checks if any of the spots to be checked have a “BLANK” symbol in them, if so, “false” is returned. Otherwise, the ints passed to the function are used as indexes for the tBoard’s vector, if the index’s contain all the same symbols, “true” is returned, if not then “false” is returned.